

AWBid Type2 Fetch Brand Safety

<go/awbid-type2-fetch-brand-safety>

fanniexu@, dominickt@, santoshkc@, awbid-eng@

Status: phase 1 [launch/190114](#)

Last Update: July 10, 2017

Motivation	2
Background	2
Overview	3
Challenges	4
Goals	4
High-level Proposal	5
Detailed Design	6
AWBid Misrepresentation Url Info	6
Bow	8
SuperMixer	8
DocTargeting Server	9
Illustrated AWBid Fetch PSI Example	9
Cat2Mixer	10
AdGroup Server	11
Cat2 Indexing	11
Alternative Considered	12
Option 1. No extra AGS call when no negative placement	12
Option 2. Only build the additional tokens for ads with negative placements	12
Logging	12
Monitoring	13
Testing Plan	13
Work Estimate	14
Review	14
Launch Summary	14
Document History	15
Appendix I. Resource estimate for PSI	16

Motivation

This project aims to increase brand safety, and it is part of the Brand Safety Code Orange Response Plan.

Background

Awbid allows AdWords remarketing campaigns to buy inventory on third-party ad exchanges. Some of these third-party exchange publishers use a placeholder URL (instead of their real URL) at bid time, which we call **domain misrepresentation**. AWBid manual review showed that 10% of our top 1200 domains had some kind of misrepresentation. Due to such publisher domain misrepresentation, the system might fail to make the right decisions regarding brand safety at bid time.

There are many projects addressing brand safety issues for AWBid:

- **Manual sweep:** manually review URLs, and filter suspicious inventory at bid time in BOW. This is a short-term solution.
- **AWBid anonymous inventory control:** consume a signal from the exchange indicating a misrepresented domain; filter based on that signal at bid time in BOW.
- **AWBid pre-bid filtering:** generalize the DMCA domain misrepresentation filter to all domains & restrict AWBid bidding via RTAS for the misrepresenting inventory. This will disallow bids from AdWords at bid time by policy enforcement.

Commented [1]: How will these projects impact the number of queries that will benefit from the work in this project?

This project addresses the issue of brand safety during the second pass of AWBid Type2 (ad fetch pass).

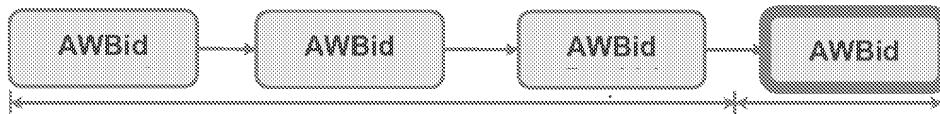


Figure 1. AWBid Brand Safety Life Cycle

TL;DR: the next three sections examines the current state of brand safety and challenges, and states the design goals for this project. Please feel free to skip ahead to High Level Proposal.

Overview

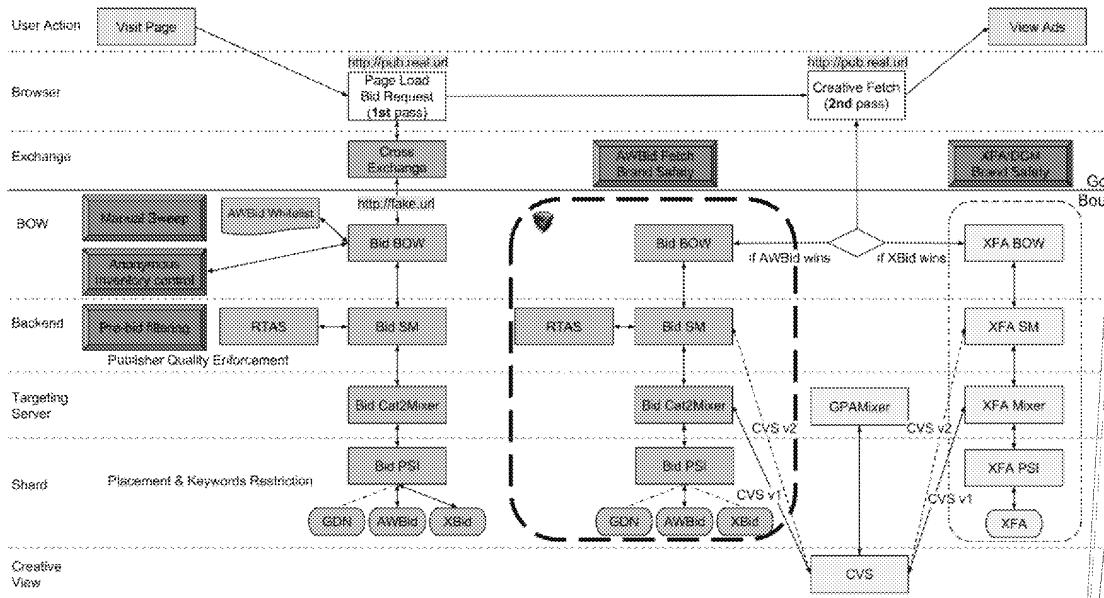


Figure 2. Bid to Fetch-ads Brand Safety in Type2 Serving Overview (Expand, Source)

XFA/DCM brand safety extracts the real URL at fetch-time in order to decide whether the bid-time domain was misrepresented. If so, and fetch-time brand safety is violated, it will show a 1x1 blank ad or a non-profit ad.

This work is similar to the XFA brand safety project, and returns blank ads when brand safety is violated at fetch-time.

When fetch-time validation fails, XFA advertisers eat the payout cost to the publisher as they charge by CPM; Google will eat the cost in the AWBid case because AWBid charges by CPC. We estimate the loss to Google will be small. Manual review shows 10% of cross exchange queries have misrepresented domains, and we expect only small fraction of that 10% will fail the brand safety check.

Note that such fetch-time validation is only suitable Type2 serving. This is because AWBid can only directly interact with the browser at fetch-time. Cross-exchange on DCM and AdWords both use Type2 serving, and are therefore suitable for this post-bid filtering at fetch-time. Type1 (e.g. Native) serving, which has no fetch pass, needs to rely on pre-bid filtering at RTAS at bid-time. We expect pre-bid filtering will be more stringent because it filters all suspicious domains. This project only filters the domains which actually fail brand safety validation.

There are many types of brand safety validation at bid-time, including:

Commented [2]: Once we are able to extract real URLs at fetch time we will have some knowledge about bid URLs that are likely to be misrepresented. Have we considered only selecting ads without page-level restrictions at bid time for those URLs to avoid the potential loss due to fetch-time validation failures?

Commented [3]: That's pre-bid filtering at RTAS.

Yet, we expect this project is more correct for the brand safety detection and can hold some revenue for AWBid.

Misrepresented domain is some general domain at bid time, e.g.: spotify.com. Even we know the pattern of the misrepresentation, we don't know whether the request page really fails until view time (e.g.: good.pub.com and bad.pub.com hide their real url as spotify.com)

Commented [4]: I think Xiangfei is suggesting something more granular than pre-bid filtering in RTAS.

That would disable all bids (or per demand segment).

Commented [5]: I think almost all the adgroups have exclusion on site, placement, keywords, vertical, etc (I should run Cat2BT to check). If it is true, this idea on misrepresented domain means we don't serve ads anyway.

Commented [6]: That is probably not true. I am curious about cat2bt requests too. In the meantime, our small-traffic experiment shows ~50% revenue impact when excluding adgroups with page-level exclusions.

Commented [7]: Hey Xiangfei, is it possible you could help confirm the experiment you mentioned above is this thread [1]?

Per offline chat with Santos, we think this experiment doesn't count placement and site restriction but only ...

Commented [8]: BTW, the PSI exclusion we care most is mentioned below in Page 10 option 2: https://docs.google.com/document/d/1g4LKw2qghO4IFYfs_KQjTTEyobeBCq_NaQ0SxrjKEyl/edit#bookmark-id.ps4mtuv7549q

Commented [9]: Regd. the original question of xiangfei - We did consider the idea of restricting bidding of "suspected urls/requests" to only those ads that don't have exclusions, but didn't prioritize yet due to lack of cycles. The data from domain misrepresentation filter ...

Commented [10]: Are you planning to log the fetch-time detected URL somewhere, e.g. QEM? Note that the current RTAS misrepresentation filter for AwBid that we are developing trains on XBID data, but with the top-URL being available on AwBid in the future w ...

Commented [11]: yes. We are still discussing whether to replace.
Original plan is:
QEM.content_url: fetch url
QEM.content_query_info.bid_url: original bid_url

- **Placement and keyword restrictions** are manually input by AdWords advertisers in order to exclude their ads from serving on specific domains, verticals, and keywords.
- **Publisher policy enforcement** prevents buying on publishers that violate AdWords policies, for reasons including copyright (e.g. piracy), legal (e.g. child porn), privacy (e.g. tripwire), content (e.g. violence), behavior (e.g. ad placement), and traffic (e.g. self-clicking).

This project aims to rerun these checks at fetch-time in addition to bid-time.

Challenges

The main challenge is to bring placement and keywords restriction to the AWBid Type2 fetch pass:

- **Fetch-time PSI:** The fetch pass does not have PSI matching. Advertiser-specified restrictions are implemented as PSI tokens. While we need much less information at fetch-time than at bid-time (geo, targeting information, etc is not needed), we do still need to create another PSI arm and fetch-time MDU to allow us to run restriction checks on the new information only available at fetch-time.
- **Fetch-time URL and page parsing:** AWBid uses only (light-weight) cross exchange doc targeting (XDT) for keywords and URL parsing at bid-time because AWBid is userlist targeted and latency-tight. At fetch-time, we need to make sure XDT (and Crawl) are available for use with the fetch-pass domain. The same is true for the RTAS page url.
- **Latency and resource challenges:** An additional PSI arm in Cat2 Indexing will potentially increase bid-time matching latency, and will require additional resources. Effort must be made to affect the AWBid shard without adversely affecting the GDN shard.
- **Ongoing CoRay Integration:** CreativeView-Server (a.k.a. CVS) decouples creative serving from matching by introducing a new server. We need to implement a solution that behaves well with the multiple CVS launch phases.

Goals

- Enforce brand safety at fetch-time. Return a blank ad when the actual publisher doesn't meet brand safety requirements.
- The highest priority is to enforce url/site level restriction.

Commented [12]: How will the bid-time and view-time behaviors for the relevant features be kept consistent?

How will this be future-proofed against new targeting features that may be added for brand safety, or launches in site/placement/etc exclusions?

Commented [13]: We propose a new arm in PSI, i.e.: https://docs.google.com/drawings/d/1wnxe_vMbGz0vIN7bwavT3kempvCAvowA_VJNgEKJcg/edit

For a new targeting type, e.g.: auto-targeting, doppeganger, we (manual-target arm, auto-targeting arm, doppeganger arm and awbid fetch arm) share the same exclusion (i.e.: filter disjunction)

For launches in site/placement/etc exclusions, I expect awbid fetch arm will go with them.

It can also save some memory in compiled query pe

Commented [14]: Ah gotcha didn't realize that an arm meant that it was shared across different queries. On the request side will the match/mdu-generation logic also be shared between view and bid time -

Commented [15]: I think almost all the adgroups have exclusion on site, placement, keywords, vertical, etc (I should run Cat2BT to check). If it is true, this idea on misrepresented domain means we don't serve ads anyway.

Commented [16]: Was this comment intended to be for the thread w/ Xiangfei, about not selecting ads w/ exclusions?

(https://docs.google.com/a/google.com/document/d/1g4LKw2qqhO4FYfs_KQjTTEy0beBCq_NaQ0SxrjKEY...)

Commented [17]: ah... yes, let me copy that to above....

Commented [18]: for consistency issue, how can we define the inconsistency?

I'd define below 2 cases:
1. data center inconsistency due to the update, e.g.: https://docs.google.com/a/google.com/document/d/1g4LKw2qqhO4FYfs_KQjTTEy0beBCq_NaQ0SxrjKEY...

Commented [19]: I'm asking about consistency on the code level, not the data level, which your 1. and 2. seem to address.

Given fixed set of advertiser settings, and fixed doc

Commented [20]: Yay, you are right that PSI should be consistent.

For mdu, we only use NegativeKeywordDocBuilder (keyword exclusion list) and FilterDocBuilder (site,

Commented [21]: Just rephrasing what fannie said earlier:

yes, the plan is to share the code in cat2 build (for consistent psi query generation) and cat2 mixer (for consistent mdu generation).

High-level Proposal

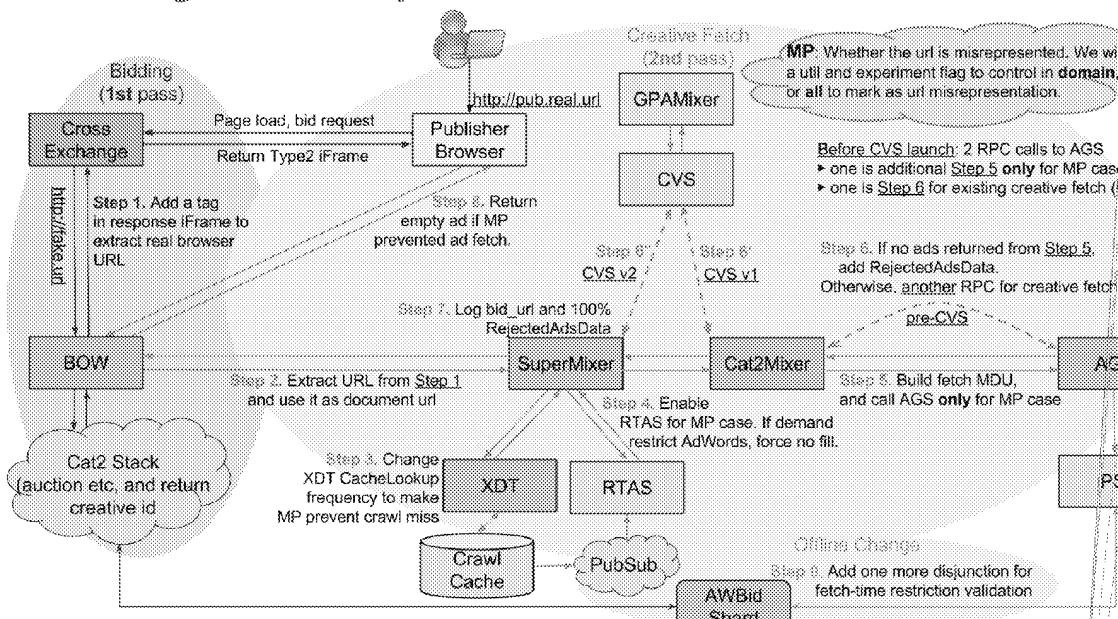


Figure 3. Bid to Fetch-ads High-level Design (Expand, Source)

The above diagram outlines the high-level system changes. Basically, we propose to implement:

1. **Misrepresented domain detection:** we will capture the real url at fetch-time, decide whether the bid-time domain was misrepresented, and use the correct url in the ContentAdRequest so that XDT and RTAS make the right brand safety decision.
2. **2 RPCs to AdGroup Server:** The new call will implement the fetch-time validation pass with a lightweight fetch-time MDU. The other call is the existing creative-fetch call (to AGS or CVS).
3. **One more arm in AWBid shard PSI Query:** We need to add an arm to existing AWBid PSI queries to support fetch-time restriction checks. An example is illustrated below. Note that this is only limited in AWBid shard with a hack (like what APPY does) before go/cat2-squery-template-builder launch. Appendix I shows how we estimate a need for 760 additional cores and increase latency by 0.1ms on average and 0.3ms in the tail. Note that it is for loadtest, and we expect the actual latency increase will be much smaller.
4. **Make XDT caller configurable:** Keyword restriction relies on Cross-Exchange Doc Targeting (XDT). We will change go/skyray-xdt-cache-simulator from being static (by experiment) to dynamic per-query so that we can increase the Crawl coverage in the case of misrepresented domains.

Commented [22]: Just to confirm: XDT & RTAS are invoked only when the real url captured at fetch time is different from the url used at bid time, right?

Commented [23]: yes

Commented [24]: Actually no now...

Per Santosh [1], we found the doc coverage is 80% in fetch-ads time. We are thinking:

- * 100% call RTAS
- * change XDT config when misrepresentation happens in *page* level
- * call AGS PSI validation when misrepresentation happens in *domain* level

[1] http://g/awbid-eng/VctHWc9xH_g/LEba7uYeBQAJ

Commented [25]: Actually for RTAS, it is already 100% RPC called [*RtasLookupProducer*], but we don't take enforcement [*PublisherPolicyProducer*] now. We propose to take enforcement for fetch-ads item brand safety.

Commented [26]: if domain is mis-represented, the bid time targeting signals can be off. Do we want to re-run the ads matching logic w/ full MDU as in bid time, to make sure advertiser targeting+restricts+filtering are matched?

Commented [27]: do we want to check if the ad matches all signals from the view time url? If so - we can probably just re-use the bid time PSI query

Commented [28]: This will have resource implications on the Crawl backend. Do you have an estimate?

Also, even without frequency optimization the URL might be new to Crawl/DocTargeting and we won't be able to return content for the request. In that case we will drop the ad if it has any page-level constraint, is that right?

Commented [29]: Do you have reference for us to estimate the resource?

"the URL might be new to Crawl/DocTargeting and we won't be able to return content for the request." -- I think so. We will get some data after step 1 bow change is release. we can extract some top misrepresentation domain cases. Given some domains, is it possible your team could tell us how frequent these domains are missed in XDT?

Commented [30]: +brohan@google.com who will come up with a more detailed doc on crawl resource estimation, but in general resources are needed in three areas:

- Crawl and processing the additional URLs (Core+RAM)
- Store the content of the URLs in the backend (Disk+potentially spindles)
- Serve the URL signals (SSD+RAM).

The last one is probably the most concerning one here...

Detailed Design

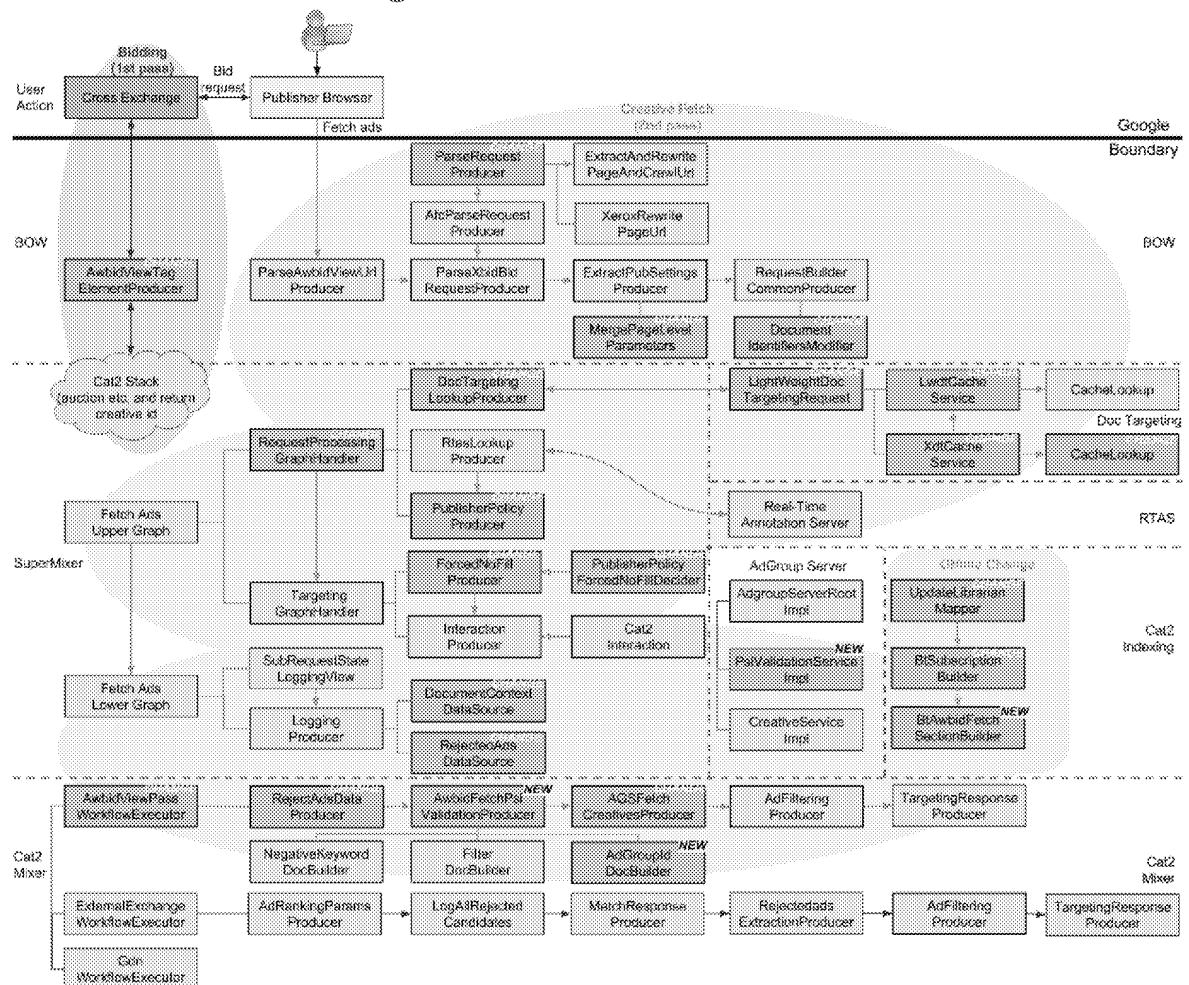


Figure 4. Bid to Fetch-ads Detailed Design (Expand, Source)

AWBid Misrepresentation Url Info

Note: below part is still changing. Per AdSense design review, we will separate into 2 phases for the proto change.

We propose the following change to ContentAdRequest (DAPI review doc).

```
    }
```

```
}
```

```
}
```

```
}
```

```
    }
```

```
}
```

Commented [31]: Where will that be called?
If we call this in Bow, can we just replace the 'bid_url' with the 'view_url' and not change much downstream (though we'd still need to actually enforce policy / brand controls at view time)?

Commented [32]: Said differently, do we really need both URLs downstream, or do we just need the one we selected with this method?

Commented [33]: I think this util will be called in bow, supermixer and cat2mixer.

For bow, we will need more functionality e.g.: check whether params.rfi is valid (e.g.: has content, http or https request, etc), rewrite the &rfi per crawl util.

We need 2URLs in downstream mainly for logging purpose, joining ads.AdQueries is really slow (Adfetch is repeated event id, and only getting 10 joining records on AdQueries and Adfetch will be timeout)

Commented [34]: We should try to find another name for this URL.

Bow

Commented [35]: As discussed offline, I imagine this URL parameter could be pretty long (it's a URL itself). Will that make the creative fetch URL too long (since we have a 2083 chars limit) and break creative fetch?

Commented [36]: We don't care the body size per offline confirm from Bow.

<http://g/awbid-eng/Bs2hzfNIHk/6uwRoXuqBgAJ>

Commented [37]: I don't think we have that 2083 character limit any more:

<https://docs.google.com/document/d/1GGTpcwKgyTA8gZjBt81xg2xy5xKogqGuAZC8DZAlk/edit#heading-cjtwnbs9t5o7>

I think BOW also moved to the 16K value of that flag. Older browsers (IEs) can still have that limit, but revenue coming from old IEs is fairly low for us to not care.

Commented [38]: +xiangfei@google.com
+aabdou@google.com
+ryanh@google.com
+jasonh@google.com

I wonder if that better belongs in DocumentContext - as opposed to being specific to AwbidViewRequest.

Commented [39]: +swise@google.com fyi

Commented [40]: (If we go for the later, at bid time, ...)

Commented [41]: Do we need both types of URLs if ...

Commented [42]: We need 2 URLs downstream

Commented [43]: Apart from the logging usecase

Commented [44]: So it really sounds like there will b

Commented [45]: Hi Edouard,

Commented [46]: Throwing some wrenches int he

Commented [47]: My 2 cents: I consider DTS a

Commented [48]: @Fannie: yes (was my proposal)

Commented [49]: Sorry - realized some of the last

Commented [50]: Last note: BOW #3 does say that

Commented [51]: in which case having a separate

Commented [52]: Having two DocumentIdentifiers

Commented [53]: The motivation behind the original

Commented [54]: IIUC, the assumption here is that

Commented [55]: So I would still advise to introduce

Commented [56]: Why do we need a whole instance

Commented [57]: @Abdel: good question. I actually

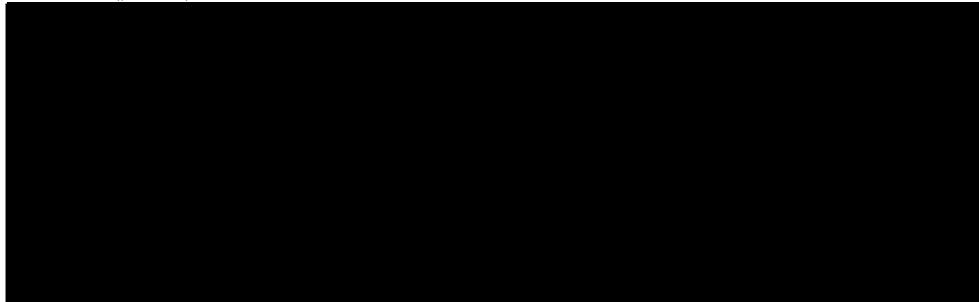
Commented [58]: @Edouard -

Commented [59]: Yes, my concern is (1) - in

SuperMixer



DocTargeting Server



Illustrated AWBid Fetch PSI Example

All of the above changes are dependencies of the PSI change. In brief, we need to check negative restriction at fetch-time, using AdGroupServer as a policy checking service. To do this, we add a branch to the AWBid PSI tree so that we can return a match if-and-only-if the fetch-time domain information satisfies the restrictions set by the creative we selected at bid-time.

Commented [60]: Add a test section to validate that the new sqquery does not alter the original psi query, via repository query-diff tool.

Commented [61]: Thanks! Added to test plan.

* OR

```
* AND field.manual_target.explanation_id:0
* AND field.target
* OR field.restrict
* NOT
  * OR field.filter keywords exclusion
* NOT
  * OR field.filter site & vertical exclusion
```

Original PSL QUERY

Bid MDU

Same negative placement list
share memory in compiled query

AND

```
(OR field.restrict ad_group_id:xxx) From awbid_c docid in AWBid Type2 Fetch
NOT (OR field.filter keywords_exclusion_list)
NOT (OR field.filter vertical_site_exclusion_list)
```

NEW

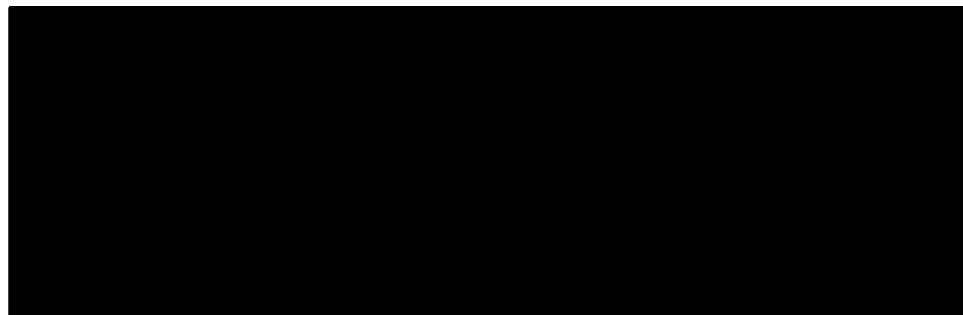
Fetch MDU

Figure 5. AWBid Shard PSI Query Example (Source, Expand)

Cat2Mixer

Commented [62]: AdGroupId and Global Doc ID are different things. Better clarify this is GlobalDocID.
Commented [63]: okay. +dominickt@google.com
Commented [64]: Just to clarify, this is more than a name change. Global doc id is PSI implementation specific, while ad_group_id is known as an id assigned in F1. We should use the latter and fix all the naming.

Commented [65]: +karimnosseir@google.com
Per our sync, you mentioned Cat2Mixer is single-thread, but multi-thread for RPC call to AGS. What else we should add in this case?
Assigned to Karim Nosseir



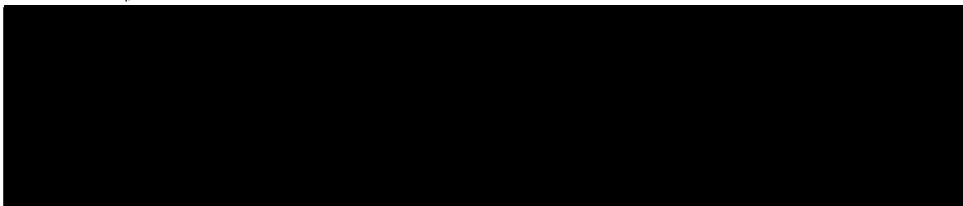
Commented [66]: Why do we allow exchanges to misrepresent content in the first place?



Commented [67]: +kenchen@google.com
+jianch@google.com
+fongshen@google.com

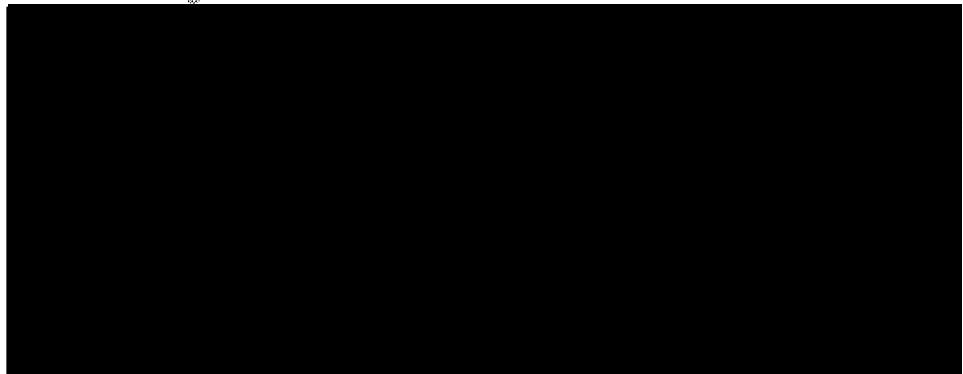
[1] <http://google3/contentads/felix/proto/creative-service.proto>?q=repeated&l=15,26,42&rcl=152781141
Assigned to Ken Chen.

AdGroup Server



Commented [68]: I think it's better we add a new request params to handle the view time validation case, current retrieval only is not exactly doing just matching check only. Or even better a new RPC interface - make it easier to monitor latency/error rate, query cost etc.

Cat2 Indexing



Commented [69]: We can also consider having PSI root send only the match request to a given shard where this ad is

Commented [70]: This is done in a really awkward way for the appy corpus today. What should have been done was allow each corpus to have its own override for initializing the subscription builder. UpdateLibrarianMapper should not have any corpus specific logic in it.

Alternative Considered

We are requesting 760 cores because the additional arm in AWBID PSI can increase bid latency. Below are some ideas copied from an [email discussion](#).

Option 1. No extra AGS call when no negative placement

If the winning ad candidate does not have negative placements, we do not need to do an extra AGS call. This requires passing a bit from bid-time to fetch-time.

What we can do is populate the "has_site_exclusion" bits in cat2 indexing pipeline (based on cat2 bt column "excluded_sites"), in serving set the bit in AdCandidate and pass them from bid to fetch flow.

Option 2. Only build the additional tokens for ads with negative placements

In BtSubscriptionBuilder, when there is NO exclusion label in sp_map, we can skip generating the negative placement, but still keep adgroup_id=xxx, otherwise the fetch-time validation will be failed. Exclusion columns include:

Logging

Commented [71]: Do you only care about the negatives? Does this work only because you're limited to remarketing? How does it play out if you add in things like keyword or other contextual signals for awbid in the future and those no longer match once we know the real page?

Commented [72]: +hangman@google.com +jieh@google.com

Hi Charles, Jie,
Do you happen to know which column corresponding to filter token
(a) url/domain (b) vertical/categories (c) keywords

I can change the sample change cl/152606326 to re-estimate the resource

Commented [73]: cc +teuijin@google.com
Hadn't actually looked at the criteria being used until now - aren't content label exclusions critical for brand safety? These are the criteria most directly associated w/ brand safety classifications and I thought part of the code orange response involved changing defaults so ...

Commented [74]: You propose a good point on content_label :)

This is optional idea on PSI, and I think we plan to use all the exclusion in PSI. Yet for MDU part, we need to make sure whether NegativeKeywordDocBuilder and FilterDocBuilder already cover content_label

Commented [75]: My understanding from the brand safety discussions is that the top priority is making sure the placement/url exclusions work and improving the coverage of the other exclusions is secondary to that (but still important and we'll get to it later).

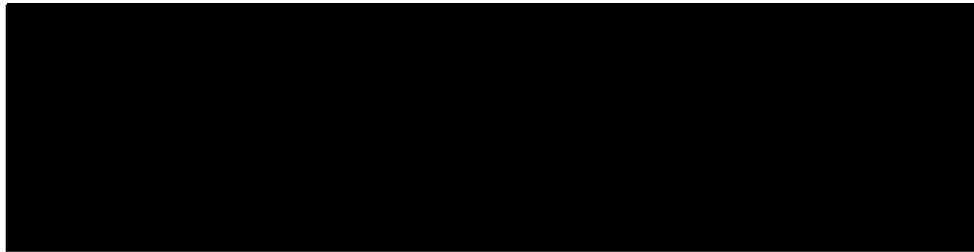
Commented [76]: Placement/url prioritization makes sense. Would you want to be able to launch that and keywords/categories independently (e.g. in case of impact on margin, etc) - and potentially same for content label exclusions?

Commented [77]: +dominickt@google.com FYI.

Per your spirit on launch site and keywords, content_label separately, we can add different experiment flags to the doc builders, and only when is_awbid_view_request && new_exp, can the filter dc ...

Commented [78]: Fannie/Dom, please look into how much additional work it'll take to support the other exclusions and whether it can be done independently later. We don't want to delay the placement work, so if it takes more than say 1-2 weeks to get the other exclusions done but the other exclusions are easy to ...

Commented [79]: Is this global docid, or adgroup id -- there isn't a "global adgroup id" concept (or... adgroup id is global, but is distinct from global docid -- sounds like someone maybe mashed up pieces and ended up with a confusing name?)



Monitoring

Add stats to Cat2Mixer to get

- Number of misrepresented domain cases per exchange
- Number of times ads are rejected due to PSI validation failure per exchange

Add stats to SuperMixer to get

- Number of times filtered by RTAS per exchange. (# of blacklist + # of demand restrict on AdWords)

Add stats to AdGroupServer new service to get

- Latency etc like current [CreativeServiceImpl](#).

Testing Plan

Bow regtest:

- [\[Done\]](#) Test new tag "&rl" is added
- Test ContentAdRequest has new fields

Global test:

- Test view-time mdu <-> PSI works, i.e.: add a exclude token to Cat2BT, and verify the fetch ads request did filter, and verify the log value. [Sample cl](#) for global test on PSI.

Query-diff:

- Validate that the new sqquery does not alter the original psi query, via [repository query-diff tool](#)

Work Estimate

- 1 month for design review and pange review.
- 1.5 month x 2 eng for development including global test.
- 1 month for experiment.

Commented [80]: +edouard@google.com

I think we should be making a decision here that is consistent with how we want this to change semantics of the serving time URL.

This has similar issues of changing semantics of the ...

Commented [81]: smartass-flogs "may" read content_url. It's not clear if it's actually tmp-AdFetchQueriers::content_url is actually used by flog: ...

Commented [82]: It seems like rtsf domain misrepresentation filter needs "url_in_creative_fetch_request" to be logged in any o ...

Commented [83]: Does RTSF consume partner logs? (I thought FE logs are mostly used for debugging).

Commented [84]: We are using RTAS and not RTSF for the misrepresentation filter, go/rtas. There isn't a huge difference from the product POV, but better use ...

Commented [85]: Does RTAS really need the full url_in_creative_fetch? Or does it just need its domain or maybe even just whether it's different from the bid: ...

Commented [86]: I think current implementation checks if domains match or not (<https://cs.corp.google.com/piper///depot/google3/ads> ...)

Commented [87]: Currently RTAS only needs the TLD (top-level domain) of the fetch URL (we are comparing TLDs when determining misrepresentation), but loggi: ...

Commented [88]: Had a quick chat with Santosh.

IIUC, there might be uses-case for consuming: ...

Commented [89]: Can anyone tell me where I can get 1/ 2/ and 3/ in Edouard's summary in the last comment on this thread?

Commented [90]: 1/ The url in bid request provided by exchange, as you already know. It is logged in ads.AdQueries.content_url

Commented [91]: +yuanzha@google.com

Fannie, it is my understanding that when the ad-tag is ...

Commented [92]: 1. +jfg@google.com above comment is for ssorigin to get top url. I don't remember your suggestion on non-webkit. Could you help

Commented [93]: For non-webkit, we'll still get a higher URL than what is available on the referer header, although it may not always be the top page

Commented [94]: Also add stats for all misrepresentation locations?

Commented [95]: Bow one is similar to cat2mixer first one. or we can directly add to BOW. For ratio (# of psi failure) / (# of misrepresented), what do you think?

Commented [96]: more monitoring > less monitoring

Review

- AdSense design review: [note](#)
- DAPI review: [note](#)

Launch Summary

15% Rasta: -0.46% impression, -0.6% clicks, -0.21% revenue Exchange*Adwords; neutral

Exchange*DBM stats.

The drop of 0.46% impression is because we filter the brand violation request in fetch-ads time.

Our monitoring also shows similar trend.



Document History

Date	Author	Description
2017-04-20	fannixu	Draft the whole design
2017-04-20	dominickt	English review, summarized Cat2Mixer changes
2017-04-25	fannixu	Add more details on PQ ForceNoFillDecider and RejectAdsData for response
2017-05-09	fannixu	Add AdGroupServer detail change after AdSense design review.

Appendix I. Resource estimate for PSI

The resource need is around **~760 cores** in total, which is likely an overestimate because we used a load test. [Resource ticket](#)

Summary

Below is the summary of the source we estimate from build single shard and Adgroup server Leaf loadtest for bid pass, and the resource for fetch-ads can be ignored (explained as below details).

Build Single Shard	mallocz-mem-in-use-by-application	+2.55%	0.22GB
	memory-usage	+0.94%	0.17GB
	mallocz-physical-memory-used	+2.42%	0.15GB
AGS Leaf loadtest (SLL)	avg_cpu_latency (%)	+1%	Commented [97]: 1% CPU on Gdn instance or Awbid instance? Note that 1% of GDN is 3,000 cores. Also add a separate table and measure view-validation CPU cost on adgroup-server instance. This is an extra RPC in addition to bid time retrieval / scoring. Commented [98]: 1% on Awbid instance only. We propose to change AWBid shard with a hack after sync with Cat2Indexing (*GDN shard has NO effect*)
	avg_server_latency (msec)	-	
	server_latency_99_percentile (msec)	-	

** Based on [go/safs/cads-ram-console](#), we further translate it as cores.

Details

Bid resource

AWBID single shard	run1 link	run2 link	run3 link	run4 link	run5
mallocz-mem-in-use-by-application	+1.82%	+13.89%	+2.49%	+3.12%	+2.7
memory-usage	+0.62%	+4.46%	+1.06%	+1.04%	+1.01%
mallocz-physical-memory-used	+1.48%	+13.54%	+3.44%	+2.66%	+2.07%
AWBID AGS Leaf loadtest	Pass link	Pass link	Pass link	Pass link	Pass link
avg_cpu_latency (%)	0.66 +/-0.298%	0.8235 +/-0.1412%	0.8011 +/-0.4039%	0.86 +/-0.2637%	0.8768 +/-0.2399%
avg_server_latency (msec)	0.03	0.05	0.018	0.05	0.062
server_latency_99_percentile (msec)	0.29	0.22	-0.04	0.29	0.39

Fetch-ads resource

Based on monitoring, there are **10K QPS** adfetch request. As we know every 100QPS needs 1 core in AdGroupServer. Given there are 10% misrepresentation domain, we are asking **10 cores** for view time.